Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 7, Part 2
Pseudorandom functions (PRF)
Pseudorandom permutations (PRP)
Random Function

- When we talk about a random function $f$, we mean
  (A) Choosing $f$ uniformly at random (and then fixing it) or
  (B) Interacting with $f$

- In particular, once we choose $f$ there is no more randomness involved

- i.e. if we query $f$ on the same input twice, we get the same result
Choosing a Uniform Function

\[ f(x) = \begin{array}{c|c}
000 & 010 \\
001 & 100 \\
010 & 100 \\
011 & 111 \\
100 & 001 \\
101 & 010 \\
110 & 010 \\
111 & 000 \\
\end{array} \]

- \( \mathcal{F}_n = \) all functions mapping \( \{0, 1\}^n \) to \( \{0, 1\}^n \)

- How big is \( \mathcal{F}_n \)?
  - Can represent a function in \( \mathcal{F}_n \) using \( n2^n \) bits
  - \( \implies |\mathcal{F}_n| = 2^{n2^n} \)

- \( n = 3 \implies \# \) of entries: \( 2^3 = 8 \)
Exercise

How many functions mapping \(\{0, 1\}^n\) to \(\{0, 1\}^m\)?

1. \(m2^n\)
2. \(2^{n2^m}\)
3. \(m2^{n2^n}\)
4. \(2^{m2^n}\)
Exercise

How many functions mapping \(\{0, 1\}^n\) to \(\{0, 1\}^m\) ?

1. \(m2^n\)
2. \(2^{n2^m}\)
3. \(m2^{n2^n}\)
4. \(2^{m2^n}\) ←
## Choosing a Uniform Function

<table>
<thead>
<tr>
<th>Method A</th>
<th>Choose uniform $f \in \mathcal{F}_n$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Method B</th>
</tr>
</thead>
<tbody>
<tr>
<td>- For each $x \in {0, 1}^n$, choose $f(x)$ uniformly in ${0, 1}^n$</td>
</tr>
<tr>
<td>- i.e. fill up the function table with uniform values</td>
</tr>
<tr>
<td>- Can view this as being done <em>on-the-fly</em>, as values are needed</td>
</tr>
</tbody>
</table>
Pseudorandom Functions (PRF)

- PRF generalizes the notion of PRG
- Instead of random-looking strings we have random-looking functions
Pseudorandom Functions (PRF)

Informal

A pseudorandom function looks like a random (i.e. uniform) function

- As for PRGs, makes no sense to talk about any fixed function being pseudorandom
- We look instead at functions chosen according to some distribution
- In particular, we look instead at keyed functions
Keyed Functions

### Keyed function $F_k$

- Let $F : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^*$ be an efficient, deterministic algorithm.
- Define $F_k(x) = F(k, x)$.
- The first input $k$ is called the key.
- $F$ is efficient $\implies$ $F$ can be computed in poly time given inputs $k$ and $x$. 

---
Length-preserving Keyed Functions

<table>
<thead>
<tr>
<th>Length-preserving keyed function $F_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>The function $F_k$ is <strong>length-preserving</strong> if:</td>
</tr>
<tr>
<td>➤ $F(k, x)$ only defined if $</td>
</tr>
<tr>
<td>➤ and $</td>
</tr>
<tr>
<td>➤ i.e. input/s and output of equal size</td>
</tr>
</tbody>
</table>
Uniform Keyed Functions

Choosing a uniform $F_k$

Choosing a uniform $k \in \{0, 1\}^n$ is equivalent to choosing the function $F_k : \{0, 1\}^n \rightarrow \{0, 1\}^n$

$F_k$ induces a distribution on $\mathcal{F}_n$

- $F_k$ naturally induces a distribution on functions from $\mathcal{F}_n$
- For uniform $k \in \{0, 1\}^n$, $\forall f \in \mathcal{F}_n$:

$$\Pr[f] = \begin{cases} 2^{-n} & f \in \{F_k\} \\ 0 & \text{otherwise} \end{cases}$$
The number of functions in $\mathcal{F}_n$ is $2^{n2^n}$

$\{F_k\}_{k \in \{0,1\}^n}$ is a subset of $\mathcal{F}_n$

The number of functions in $\{F_k\}_{k \in \{0,1\}^n}$ is at most $2^n$

$\{F_k\}$ contains only a tiny fraction of $\mathcal{F}_n$:

$$2^n \ll 2^{n2^n}$$
Pseudorandom Functions

Definition

$F$ is a pseudorandom function if $F_k$, for uniform key $k \in \{0, 1\}^n$, is indistinguishable from a uniform function $f \in \mathcal{F}_n$. 
Pseudorandom Functions

PRG

$D$ is given access to a bit-string

$$|\Pr_{x \leftarrow U_n}[D(G(x)) = 1] - \Pr_{y \leftarrow U_{p(n)}}[D(y) = 1]| \leq \epsilon(n)$$
## Pseudorandom Functions

### PRG

$D$ is given access to a bit-string

$$\left| \Pr_{x \leftarrow \mathcal{U}_n} [D(G(x)) = 1] - \Pr_{y \leftarrow \mathcal{U}_p(n)} [D(y) = 1] \right| \leq \epsilon(n)$$

### PRF

$D$ is given the description of $f$ or $F_k$

$$\left| \Pr_{k \leftarrow \{0,1\}^n} [D^{F_k}() = 1] - \Pr_{f \leftarrow \mathcal{F}_n} [D^{f}() = 1] \right| \leq \epsilon(n)$$
Pseudorandom Functions

**PRG**

$D$ is given access to a bit-string

$$|\Pr_{x \leftarrow U_n}[D(G(x)) = 1] - \Pr_{y \leftarrow U_{p(n)}}[D(y) = 1]| \leq \epsilon(n)$$

**PRF**

$D$ is given the description of $f$ or $F_k$

$$|\Pr_{k \leftarrow \{0, 1\}^n}[D^{F_k}(\cdot) = 1] - \Pr_{f \leftarrow \mathcal{F}_n}[D^f(\cdot) = 1]| \leq \epsilon(n)$$

**Problem**

- Description of $f$ is at least $n2^n$ bits long i.e. exponential
- $D$ has polynomial capabilities
Pseudorandom Functions

PRF

\(D\) is given the description of oracle access to \(f\) or \(F_k\)

\[|\Pr_{k \leftarrow \{0, 1\}^n} [D^{F_k}(\cdot) = 1] - \Pr_{f \leftarrow \mathcal{F}_n} [D^f(\cdot) = 1]| \leq \epsilon(n)\]

Solution

- Now \(D\) can query \(f\) (resp. \(F_k\)) at most poly times
Pseudorandom Functions (PRFs)

**Definition (refined)**

$F$ is a pseudorandom function if $F_k$, for uniform key $k \in \{0, 1\}^n$, is such that for all **poly-time distinguishers** $D$: 

$$|\Pr_{k \leftarrow \{0, 1\}^n}[D^{F_k}(\cdot) = 1] - \Pr_{f \leftarrow \mathcal{F}_n}[D^f(\cdot) = 1]| \leq \epsilon(n)$$

$D$ can query $f$ (resp. $F_k$) on any input $x$ at most **poly times**
PRF vs. RF

\[ f \in \text{Func}_n \text{ chosen uniformly at random} \]

World 0

\[ f(x_1), \ldots, f(x_t) \]

\[ \text{World 1} \]

\[ k \in \{0,1\}^n \text{ chosen uniformly at random} \]

\[ \text{poly-time} \]

\[ F_k(x_1), \ldots, F_k(x_t) \]
## PRF vs. RF

<table>
<thead>
<tr>
<th>Warning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attacker (distinguisher $D^{F_k}$) <strong>does not</strong> have access to the key $k$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Meaningless to distinguish $F_k$ from $f$ for a <strong>known key</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ Recall: $F_k(x)$ is efficiently computable for any $k, x$</td>
</tr>
<tr>
<td>▶ $D$ queries the oracle on $x$ and gets a result $y$</td>
</tr>
<tr>
<td>▶ As $D$ knows $k$ (and $x$), it computes $y' = F_k(x)$</td>
</tr>
<tr>
<td>▶ If $y' = y$ output 1; else 0</td>
</tr>
<tr>
<td>▶ $\implies$ able to distinguish with $\Pr \approx 1$</td>
</tr>
</tbody>
</table>
PRF vs. RF

Warning

Attacker (distinguisher $D^{F_k}$) does not have access to the key $k$

Meaningless to distinguish $F_k$ from $f$ for a known key

► Recall: $F_k(x)$ is efficiently computable for any $k, x$
► $D$ queries the oracle on $x$ and gets a result $y$
► As $D$ knows $k$ (and $x$), it computes $y' = F_k(x)$
► If $y' = y$ output 1; else 0
► \(\Rightarrow\) able to distinguish with $\Pr \approx 1$

Note

$f \in \mathcal{F}_n$: $\Pr[f(x) = y'] = \frac{1}{2^n}$
Is the Following PRF Secure?

\[ F_k(x) = 0^n \]
Is the Following PRF Secure?

$F_k(x) = 0^n$

**Distinguisher $D$**

1. Query $O$ on arbitrary $x$: $y = O(x)$ (note: $O = \{f, F_k\}$)
2. If $y = 0^n$ output $1$; otherwise output $0$
Is the Following PRF Secure?

\[ F_k(x) = 0^n \]

Distinguisher \( D \)

1. Query \( \mathcal{O} \) on arbitrary \( x \): \( y = \mathcal{O}(x) \) (note: \( \mathcal{O} = \{ f, F_k \} \))
2. If \( y = 0^n \) output 1; otherwise output 0

Analysis

\[
|\Pr_{k \leftarrow \{0,1\}^n}[D^{F_k}(\cdot) = 1] - \Pr_{f \leftarrow \mathcal{F}_n}[D^f(\cdot) = 1]| \\
= |1 - \frac{1}{2^n}| \approx 1 \lessapprox \text{negl}
\]
Is the Following PRF Secure?

\[ F_k(x) = k \]
Is the Following PRF Secure?

\[ F_k(x) = k \]

**Distinguisher** \( D \)

1. Query \( O \) on arbitrary \( x_1, x_2 \): \( y_1 = O(x_1), y_2 = O(x_2) \)
2. If \( y_1 = y_2 \) output 1; otherwise output 0
Is the Following PRF Secure?

\[ F_k(x) = k \]

**Distinguisher \( D \)**

1. Query \( \mathcal{O} \) on arbitrary \( x_1, x_2 \): \( y_1 = \mathcal{O}(x_1), y_2 = \mathcal{O}(x_2) \)
2. If \( y_1 = y_2 \) output 1; otherwise output 0

**Analysis**

\[
|\Pr_{k \leftarrow \{0,1\}^n} [D^{F_k} = 1] - \Pr_{f \leftarrow \mathcal{F}_n} [D^f = 1]| \\
= |1 - \frac{1}{2^n}| \approx 1 \lesssim \text{negl}
\]
Is the Following PRF Secure?

\[ F_k(x) = k \oplus x \]
Is the Following PRF Secure?

$$F_k(x) = k \oplus x$$

**Distinguisher** $D$

1. Query $\mathcal{O}$ on arbitrary $x_1, x_2$: $y_1 = \mathcal{O}(x_1), y_2 = \mathcal{O}(x_2)$
2. If $(x_1 \oplus x_2) = (y_1 \oplus y_2)$ output $1$; otherwise output $0$
Is the Following PRF Secure?

\[ F_k(x) = k \oplus x \]

Distinguisher \( D \)

1. Query \( \mathcal{O} \) on arbitrary \( x_1, x_2 \): \( y_1 = \mathcal{O}(x_1), y_2 = \mathcal{O}(x_2) \)
2. If \( (x_1 \oplus x_2) = (y_1 \oplus y_2) \) output 1; otherwise output 0

Analysis

\[
|\Pr_{k \leftarrow \{0,1\}^n}[D^{F_k}(\cdot) = 1] - \Pr_{f \leftarrow \mathcal{F}_n}[D^{f}(\cdot) = 1]| \\
= |1 - \frac{1}{2^n}| \approx 1 \ll \text{negl}
\]
Is the Following PRF Secure?

\[ O = F_k \]

\[ \Pr[x_1 \oplus x_2 = f(x_1) \oplus f(x_2)] = 1 \]

\[ O = f \]

\[ \Pr[x_1 \oplus x_2 = f(x_1) \oplus f(x_2)] = \]

\[ \Pr[f(x_2) = x_1 \oplus x_2 \oplus f(x_1)] = \frac{1}{2^n} \]
PRFs vs. PRGs

PRF implies PRG

PRF $F$ immediately implies PRG $G$:

- Define $G(k) = F_k(0 \ldots 0) | F_k(0 \ldots 1)$

- i.e. $G(k) = F_k(0^n) | F_k(1^n) | F_k(2^n) | \ldots$
  where $i_n$ denotes the $n$-bit encoding of $i$

- Try to prove it formally (exercise 3.14).

PRF is a PRG with random access

PRF can be viewed as a PRG with random access to exponentially long output:

- The function $F_k$ can be viewed as the $n2^n$-bit string $F_k(0 \ldots 0) | \ldots | F_k(1 \ldots 1)$
Permutations

Permutation

- Let \( f \in \mathcal{F}_n \)
- \( f \) is a permutation if it is a bijection
  - This means that the inverse \( f^{-1} \) exists
- Let \( \mathcal{P}_n \subset \mathcal{F}_n \) be the set of permutations
- What is \( |\mathcal{P}_n| \)?

\[
|\mathcal{P}_n| = 2^n!
\]
Let $F$ be a length-preserving, keyed function

$F$ is a keyed permutation if

1. $F_k$ is a permutation for every $k$ and
2. $F_k^{-1}$, the inverse of $F_k$, is efficiently computable
Pseudorandom Permutations (PRPs)

<table>
<thead>
<tr>
<th>Pseudorandom Permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>▶ $F$ is a <strong>pseudorandom permutation</strong> if $F_k$, for uniform key $k \in {0, 1}^n$, is indistinguishable from a uniform permutation $f \in \mathcal{P}_n$</td>
</tr>
<tr>
<td>▶ Even if attacker can query the function <strong>and its inverse</strong></td>
</tr>
</tbody>
</table>
PRP is Indistinguishable from PRF

Fact

A random permutation is indistinguishable from a random function for large enough $n$

$\implies$ in practice, PRPs are also good PRFs
Do PRFs/PRPs exist?

- PRF is a stronger primitive than PRG
  - PRF $\Rightarrow$ PRG
- We don’t know if PRGs exist
- $\Rightarrow$ we don’t know if PRFs exist

In practise

- Stream ciphers $\Rightarrow$ PRGs
- Block ciphers $\Rightarrow$ PRPs/PRFs

Next lecture

CPA-secure encryption using PRF/PRP
End

Reference: Section 3.5.1